

Application No. 09/595776 (Docket: MIPS.0166-00-US)
37 CFR 1.111 Amendment dated 03/14/2006
Reply to Office Action of 12/14/2005

REMARKS/ARGUMENTS

In the Office Action, the Examiner noted that claims 1-20 are pending in the application. The Examiner additionally stated that claims 1-20 are rejected. By this amendment, claims 1, 6, and 11 have been amended. Hence, claims 1-20 are pending in the application.

Applicant hereby requests further examination and reconsideration of the application, in view of the foregoing amendments.

In the Claims

Rejections Under 35 U.S.C. §112

The Examiner rejected claims 6-10 and 17 under 35 U. S.C. 1 12, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Regarding claim 6, the Examiner noted that the last paragraph recites the limitation "said fetch algorithm," and that there is insufficient antecedent basis for this limitation.

In response, Applicant has amended independent claim 6 to recite "said fetch stage," as is recited earlier in the claim. In view of the noted amendment to independent claim 6, Applicant respectfully requests that the rejections of claims 6-10 and 17 be withdrawn.

Rejections Under 35 U.S.C. §102(b)

The Examiner rejected claims 1-15 and 18-19 under 35 U.S.C. §102(b) as being anticipated by Yoaz et al., "Speculative Techniques for Improving Load Related Scheduling," May 1999 (as applied in the previous Office action and herein referred to as Yoaz), in view of Hoyt et al., U.S. Patent No. 5,604,877 (herein referred to as Hoyt). In addition, the Examiner cited Parady, U.S. Patent 5,933,627 (as applied in the previous Office Action and hereinafter referred to as "Parady") as extrinsic evidence for showing that it is common to have separate hardware streams for each thread. Applicant respectfully traverses the Examiner's rejections.

Prior to providing a claim-by-claim analysis, a brief summary of the teachings of Yoaz, Hoyt, and Parady are provided below, vis-à-vis the invention disclosed by Applicant in

Application No. 09/595776 (Docket: MIPS.0166-00-US)
37 CFR 1.111 Amendment dated 03/14/2006
Reply to Office Action of 12/14/2005

the instant application. This information is provided to aid the Examiner during reconsideration of the claims.

Yoaz discloses three techniques to address *instruction scheduler* limitations in an out-of-order engine, where the *instruction scheduler* is responsible for *dispatching instructions to execution units* based upon dependencies, latencies, and resource availability. The problem that is noted by Yoaz, and which motivates his *instruction scheduler* techniques, is that dynamic latencies of load instructions are unknown, so *scheduling* dependent instructions is based on either load-use delay or pessimistic delay. (cf. Abstract) Yoaz expands his teachings in the are of hit/miss predictions by noting that the new concept presented is to predict which loads will miss the cache, thus *delaying the dependent instructions* until the needed data is fetched. Yoaz opines that this increases performance directly by saving a few clocks through *the scheduling of load-dependent instructions to execute* at the exact time the data is retrieved. Yoaz's technique involves a hardware approach that is based on a *per-load binary prediction* of a hit or miss in the cache. (cf. page 44, lines 8-27) The author specifically notes that once all instruction dependencies have been resolved, the *scheduler's* remaining task is to *dispatch instructions* in a way that maximizes the utilization of available resources, while minimizing instruction latencies. Further, Yoaz suggests dynamically predicting whether a specific load with hit or miss the cache, *to facilitate the scheduling operation*. (cf. page 46, lines 17-31) Yoaz moreover intimates that multi-threading may benefit from hit/miss prediction, and that the prediction may be used to govern a thread switch if a load is predicted to miss the L2 cache. (cf. page 47, lines 25-29)

Hoyt teaches a technique for resolving return from subroutine instructions. The instructions are resolved in four stages. A first stage predicts call subroutine instructions and return from subroutine instructions within the instruction stream. The first stage stores a return address in a return register when a call subroutine instruction is predicted. The first stage predicts a return to the return address in the return register when a return from subroutine instruction is predicted. A second stage decodes each call subroutine and return from subroutine instruction in order to maintain a return stack buffer that

Application No. 09/595776 (Docket: MIPS.0166-00-US)
37 CFR 1.111 Amendment dated 03/14/2006
Reply to Office Action of 12/14/2005

stores a stack of return addresses. When the second stage decodes a call subroutine instruction, a return address is pushed onto the return stack buffer. Correspondingly, each time the second stage decodes a return from subroutine instruction, a return address is popped off of the return stack buffer. The second stage verifies predictions made by the first stage and predicts return addresses for return from subroutine instructions that were not predicted by the first stage. A third stage executes return from subroutine instructions such that the predictions are verified. Finally, a fourth stage retires return from subroutine instructions and ensures that no instructions fetch after a mispredicted return address are committed into permanent state. (Abstract) Before an instruction is fetched, a current a current Instruction Pointer (IP) is passed to a branch target buffer circuit to learn if there is an upcoming branch instruction that directs the microprocessor to a non-sequential address. The branch target buffer circuit examines a branch target buffer cache using the instruction pointer, looking for an upcoming branch instruction. If the circuit finds an upcom branch instruction, "hit" has occurred and the branch target buffer circuit makes a branch prediction using the branch information from the cache. (col. 8, lines, 41-53)

Parady discloses a method and apparatus for switching between threads of a program *in response to a long-latency event*. In one embodiment, the long-latency events are load or store operations which trigger a thread switch *if there is a miss* in the level 2 cache. In addition to providing separate groups of registers for multiple threads, a group of program address registers pointing to different threads are provided. A switching mechanism switches between the program address registers *in response to the long-latency events*. (cf. Abstract) Parady also defines the process whereby a multithreading processor interleaves threads in such a manner as described above (i.e., in response to a long-latency event) as "coarse-grain multithreading," (cf. col. 2, lines 8-10) and furthermore teaches the concept of "a switching mechanism [that] switches between the program address registers *in response to the long-latency events*. Parady discloses the switching mechanism as "[t]hread switching logic 112 provided to give a hardware thread switching capability. The indication that a thread switch is required is provided on a line 114 providing an *L2-miss indication from cache control/system interface 22*." He further

Application No. 09/595776 (Docket: MIPS.0166-00-US)
37 CFR 1.111 Amendment dated 03/14/2006
Reply to Office Action of 12/14/2005

teaches that “[u]pon such an indication, a switch to the next thread will be performed.” (cf. col. 3, lines 57-62) Furthermore, in a background discussion of his invention, Parady cites an IBM article that distinguishes between processors to which his invention is directed (i.e., coarse-grain multithreaded processors) and fine-grain multithreaded processors, that is, those processors which interleave threads on a cycle-by-cycle basis. (cf. col. 2, lines 6-10) Parady’s invention and disclosure is directed towards problems associated with coarse-grain processors: those processor that switch threads *in response to* long latency events.

In contrast to the teachings of Yoaz, Hoyt, and Parady, Applicant’s invention is directed towards a processor having multiple hardware streams supporting multiple data threads, and a data cache. In this processor, Applicant discloses a system for *fetching* instructions from one to P of the multiple hardware streams to a pipeline, where P is less than the number of multiple hardware streams. The system has multiple hit/miss predictors, a fetch stage, and an instruction scheduler. The multiple hit/miss predictors are each associated with a corresponding one of the multiple hardware streams, and are each configured to forecast whether corresponding instructions from the corresponding one of the multiple hardware streams will hit or miss the data cache. The multiple hit/miss predictors forecast whether the corresponding instructions from the corresponding one of the multiple hardware streams will hit or miss the data cache prior to when the corresponding instructions enter into a dispatch stage in the pipeline. The fetch stage is coupled to the multiple hit/miss predictors. The fetch stage simultaneously fetches every cycle, the instructions from the one to P of the multiple hardware streams to the pipeline and furthermore selects, on a cycle-by-cycle basis, the one to P of the multiple hardware streams from which to fetch the instructions. The instruction scheduler manages access for the multiple hardware streams to a set of functional resources for processing instructions from the multiple hardware streams, wherein at any point in time, the instruction scheduler manages access for a given one of the multiple hardware streams according to a priority record, regardless of any priority associated with the multiple data threads.

Application No. 09/595776 (Docket: MIPS.0166-00-US)
37 CFR 1.111 Amendment dated 03/14/2006
Reply to Office Action of 12/14/2005

Yoaz's technique involving hit/miss prediction is clearly directed towards scheduling instructions for execution which have already been fetched into the pipeline and that have entered a dispatch stage for scheduling for execution. Such a technique is disadvantageous because the instructions have already entered the pipeline and have reached the dispatch stage. If Yoaz's hit/miss prediction results in a miss prediction, then instructions between the fetch stage and the dispatch stage must be flushed in addition to those within the fetch stage. Yoaz fails even to note the problem that is addressed by Applicant in the instant application, to wit that if the fact that an instruction will miss the data cache could be known early in the process, then the fetching of instructions that might eventually be flushed may be avoided. While it is clear that a hit/miss prediction in a dispatch stage must certainly be coupled to fetch logic in order to redirect the fetching of instructions in the event of a miss prediction, because Yoaz's technique performs the hit/miss predication at dispatch, the pipeline stages above dispatch must be flushed in the event of a miss prediction. Applicant's invention, in contrast, makes the prediction at the fetch stage itself.

It therefore does not follow that Yoaz anticipates a system for making a hit/miss prediction at a fetch stage of a multi-threaded processor pipeline, for his article is directed towards making such predictions at a dispatch stage.

In addition, Hoyt teaches caching previously executed branch target addresses and indexing those addresses according to the value of a current instruction pointer. Hoyt is entirely silent with regard to predictions on whether instructions will hit or miss a data cache. Applicant has searched Hoyt, and cannot find any reference to predicting whether instructions will be found in a data cache.

In view of the above summarizations, a claim-by-claim analysis will now be presented.

Amended claim 1 is provided below for ease of reference.

1. In a processor having multiple hardware streams supporting multiple data threads, and a data cache, a system for fetching instructions from one to P of the multiple hardware streams to a pipeline, where P is less than the number of multiple hardware streams, the system comprising:

Application No. 09/595776 (Docket: MIPS.0166-00-US)
37 CFR 1.111 Amendment dated 03/14/2006
Reply to Office Action of 12/14/2005

multiple hit/miss predictors, each associated with a corresponding one of the multiple hardware streams, said each configured to forecast whether corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss the data cache, wherein said multiple hit/miss predictors forecast whether said corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss the data cache prior to when said corresponding instructions enter into a dispatch stage in the pipeline; and

a fetch stage, coupled to said multiple hit/miss predictors, configured to simultaneously fetch every cycle, the instructions from the one to P of the multiple hardware streams to the pipeline, and configured to select, on a cycle-by-cycle basis, the one to P of the multiple hardware streams from which to fetch the instructions; and

an instruction scheduler, coupled to said fetch stage, for managing access for the multiple hardware streams to a set of functional resources for processing instructions from the multiple hardware streams, wherein at any point in time, said instruction scheduler manages access for a given one of the multiple hardware streams according to a priority record, regardless of any priority associated with the multiple data threads.

Claim 1 recites, in combination, within a processor having multiple hardware streams supporting multiple data threads, and a data cache, a system for fetching instructions from one to P of the multiple hardware streams to a pipeline. The system has multiple hit/miss predictors that are each associated with a corresponding one of the multiple hardware streams. In addition, each of the multiple hit/miss predictors is configured to forecast whether corresponding instructions from the corresponding one of the multiple hardware streams will hit or miss the data cache. The multiple hit/miss predictors forecast whether said corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss the data cache prior to when said corresponding instructions enter into a dispatch stage in the pipeline. The system

Application No. 09/595776 (Docket: MIPS.0166-00-US)
37 CFR 1.111 Amendment dated 03/14/2006
Reply to Office Action of 12/14/2005

additionally includes a fetch stage that is coupled to the multiple hit/miss predictors. The fetch stage simultaneously fetches every cycle, the instructions from the one to P of the multiple hardware streams to the pipeline, and selects, on a cycle-by-cycle basis, the one to P of the multiple hardware streams from which to fetch the instructions. The system also has an instruction scheduler that manages access for the multiple hardware streams to a set of functional resources for processing instructions from the multiple hardware streams. At any point in time, said instruction scheduler manages access for a given one of the multiple hardware streams according to a priority record, regardless of any priority associated with the multiple data threads.

In rejection of claim 1, the Examiner notes that Yoaz has taught in a processor having multiple hardware streams supporting multiple data threads, and a data cache, a system for fetching instructions from one to P of the multiple hardware streams to a pipeline, where P is less than the number of multiple hardware streams. (For purposes of the examination, the Examiner noted that P is interpreted as being equal to 1), the system comprising:

a) multiple hit/miss predictors, each associated with a corresponding one of the multiple hardware streams, said each configured to forecast whether corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss the data cache. (See page 47, column 1, and note the paragraph beginning with "Another.. .". From these citations, the Examiner pointed out that multiple threads clearly exist, and a predictor is used for each thread. Consequently, the examiner concluded that there are multiple predictors. Also, the Examiner inferred that forecasts of these predictors would be used by the fetching hardware (if predicted to hit, continue fetching from same stream; otherwise, switch and fetch from another stream).

b) a fetch stage, coupled to said multiple hit/miss predictors, configured to simultaneously fetch every cycle, the instructions from the one to P of the multiple hardware streams to the pipeline, and configured to select, on a cycle-by-cycle basis, the one to P of the multiple hardware streams from which to fetch the instructions. Again, referring to page 47, column 1, and specifically noting the paragraph beginning with

Application No. 09/595776 (Docket: MIPS.0166-00-US)
37 CFR 1.111 Amendment dated 03/14/2006
Reply to Office Action of 12/14/2005

"Another..," the examiner opined that the fetch algorithm includes the steps of fetching from a current thread as long as a miss is not predicted, and when a miss is predicted, switching to and fetching instructions from a second thread. The Examiner elaborated that it should be realized that this happens on a cycle-by-cycle basis, as instructions are executed every cycle. Whenever a load instruction appears, the prediction must be made, and if multiple loads occur in a row then the predictions are made for those consecutive cycles.

The Examiner acknowledged that Yoaz does not explicitly teach that said multiple hit/miss predictors forecast whether corresponding instructions from a corresponding one of the multiple hardware streams will hit or miss the data cache prior to when the corresponding instructions enter into a dispatch stage in the pipeline. The examiner stated that Hoyt, however, has taught the concept of making predictions for instructions before the instructions are fetched (and also before any dispatch stage). More specifically, it was noted that the address associated with the instruction (the PC) is applied to a predictor in order to make a prediction fast enough such that the results of the prediction are seen and used by the system as soon as possible. (See column 8, lines 41-53. The Examiner noted that a person of ordinary skill in the art would have recognized that a similar pre-dispatch prediction scheme would be useful in Yoaz because such a scheme would allow a load miss to be predicted as soon as possible, and consequently, a new stream would be selected as soon as possible, thereby reducing any flushing and/or delay associated with a delayed prediction. The examiner stated that the implementation of Hoyt's prediction system in Yoaz would result in predicting a load miss when it is time to fetch the load. If a load is predicted to miss while it is being fetched, then in the very next cycle a new stream may be fetched, thereby eliminating any delay and, as a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Yoaz to forecast cache-miss predictions for instructions prior to those instructions entering a dispatch stage.

Applicant respectfully disagrees with the Examiner's characterizations of both Yoaz and Hoyt, in addition to the teachings of Applicant as claimed. In response, Applicant asserts that Yoaz's teaching is restricted to making hit/miss predictions when instructions enter

Application No. 09/595776 (Docket: MIPS.0166-00-US)
37 CFR 1.111 Amendment dated 03/14/2006
Reply to Office Action of 12/14/2005

the dispatch stage of a pipeline where the instructions, having already been fetched into the pipeline and provided to the dispatch stage, are scheduled for execution.. Yoaz does not address making such predictions higher up in the pipeline, at the fetch stage, to circumvent the problems associated with flushing stages between fetch and dispatch in the event of a miss prediction. Second, Applicant is unable to find any reference to multiple hit/miss predictors in Yoaz, as is noted by the Examiner. According to Yoaz, a single predictor could be fed from the multiple streams, thus adding delay to the pipeline. Secondly, Hoyt teaches branch target address prediction through use of a branch target buffer. The branch target buffer stores an *address* of a branch target instruction. Hoyt uses the term "hit" to mean that an entry in the branch target buffer is found that corresponds to a current instruction pointer. Hoyt does not teach, suggest, allude to, or even hint that a prediction can be made to determine whether instructions from a corresponding hardware streams will hit or miss a data cache prior to when the instructions enter into a dispatch stage. Hoyt is silent with reference to multiple hardware streams, in fact. Applicant respectfully asserts that one skilled in the art would not be motivated to use the teachings of Hoyt in combination with Yoaz to provide the invention of claim 1 because Hoyt does not suggest that any type of prediction of cache hit/miss would be desirable.

In addition, neither Yoaz nor Hoyt teach an instruction scheduler for managing access for the multiple hardware streams to a set of functional resources for processing instructions from the multiple hardware streams, where at any point in time, the instruction scheduler manages access for a given one of the multiple hardware streams according to a priority record, regardless of any priority associated with the multiple data threads.

For these reasons, Applicant respectfully requests that the Examiner withdraw his rejection of claim 1.

With respect to claims 2-5 and 14-15, these claims depend from claim 1 and add further limitations that are neither anticipated nor made obvious by Yoaz, Hoyt, Parady, or any combination of the noted references. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections of claims 2-5 and 14-15.

Application No. 09/595776 (Docket: MIPS.0166-00-US)
37 CFR 1.111 Amendment dated 03/14/2006
Reply to Office Action of 12/14/2005

In a manner substantially similar to claim 1, claim 6 recites, in combination with other elements, a data cache, comprising a plurality of levels; multiple hit/miss predictors, and a fetch stage. The multiple hit/miss predictors are each associated with a corresponding one of the multiple hardware streams, and each are configured to forecast whether corresponding instructions from the corresponding one of the multiple hardware streams will hit or miss said data cache. The multiple hit/miss predictors forecast whether the corresponding instructions from the corresponding one of the multiple hardware streams will hit or miss said data cache prior to when said corresponding instructions enter into a dispatch stage in a pipeline of the processor. Moreover, claim 6 recites an instruction scheduler for managing access for the multiple hardware streams to a set of functional resources for processing instructions from the multiple hardware streams, where at any point in time, the instruction scheduler manages access for a given one of the multiple hardware streams according to a priority record, regardless of any priority associated with the multiple data threads. As noted above in traversal of the rejection of claim 1, the recited elements and limitations are entirely absent from the teachings of Yoaz because Yoaz's technique for hit/miss prediction is performed at the dispatch level on instructions that have already been provided by fetch logic to the pipeline. Furthermore, it has been argued that Hoyt teaches branch prediction and not prediction of whether an instruction will hit/miss a data cache when executed. Moreover, all of the references are silent with regard to an instruction scheduler that manages access for multiple hardware streams to a set of functional resources for processing instructions from the multiple hardware streams, where at any point in time, the instruction scheduler manages access for a given one of the multiple hardware streams according to a priority record, regardless of any priority associated with the multiple data threads. Consequently, for reasons substantially noted above in arguments presented in traversal of the Examiner's rejection of claim 1, Applicant asserts that claim 6 is allowable over the cited references and respectfully requests that the Examiner withdraw the rejection of claim 6.

With respect to claims 7-10, these claims depend from claim 6 and add further limitations that are neither anticipated nor made obvious by Yoaz, Hoyt, Parady, or any of the

Application No. 09/595776 (Docket: MIPS.0166-00-US)
37 CFR 1.111 Amendment dated 03/14/2006
Reply to Office Action of 12/14/2005

references in combination. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 7-10.

Like claims 1 and 6, claim 11 recites, a method for simultaneously fetching instructions every cycle from up to P hardware streams to a pipeline. The method includes, for each of the hardware streams, making a hit/miss prediction by a corresponding one of associated hit/miss predictors as to whether corresponding instructions for the each of the multiple hardware streams previously fetched will hit or miss the data cache, where the making of the hit/miss prediction is performed prior to when the corresponding instructions enter into a dispatch stage in the pipeline; and selecting, on a cycle-by-cycle basis, the P hardware streams from which to fetch the instructions. The method also includes managing access for the multiple hardware streams to a set of functional resources for processing instructions from the multiple hardware streams, wherein at any point in time, the managing for a given one of the multiple hardware streams is accomplished according to a priority record, regardless of any priority associated with the multiple data threads. None of the noted references suggest anything about these above noted elements, in particular that data cache hit/miss predictions are made prior to when instructions are provided by the pipeline for dispatch. Nor do any of the references teach managing access to functional resources as recited. Consequently, for reasons substantially noted above in arguments presented in traversal of the Examiner's rejections of claim 1 and claim 6, Applicant asserts with respect to the rejection of claim 11 that Yoaz teaches how to more effectively schedule instructions *that have already been fetched into a pipeline*, for execution. Applicants invention makes hit/miss predictions prior to when instructions enter the dispatch stage.

Accordingly, Applicant respectfully requests that the Examiner withdraw his rejection of claim 11.

With respect to claims 12-13 and 18-19, these claims depend from claim 11 and add further limitations that are neither anticipated nor made obvious by any combination of Yoaz, Hoyt, and Parady. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 12-13 and 18-19.

Application No. 09/595776 (Docket: MIPS.0166-00-US)
37 CFR 1.111 Amendment dated 03/14/2006
Reply to Office Action of 12/14/2005

The Examiner also rejected claims 16-17 and 20 under 35 U.S.C. §103(a) as being unpatentable over Yoaz in view of Hoyt, as applied in the rejections above discussed above, in view of Ryan, U.S. Patent No. 5,694,572. Applicant respectfully traverses and asserts that neither Yoaz nor Hoyt teaches the limitations and elements recited in claims 1, 6, or 11, as noted in arguments provided above. Accordingly, since claim 16 adds further limitations over that recited in claim 1, it is respectfully requested that the rejection of claim 16 be withdrawn. Likewise, since claim 17 adds further limitations over that recited in claim 6, it is respectfully requested that the rejection of claim 17 be withdrawn. In addition, since claim 20 adds further limitations over that recited in claim 11, it is respectfully requested that the rejection of claim 20 be withdrawn.

Application No. 09/595776 (Docket: MIPS.0166-00-US)
37 CFR 1.111 Amendment dated 03/14/2006
Reply to Office Action of 12/14/2005

CONCLUSIONS

In view of the arguments advanced above, Applicant respectfully submits that claims 1-20 are in condition for allowance. Reconsideration of the rejections is requested, and allowance of the claims is solicited.

Applicant earnestly requests that the Examiner contact the undersigned practitioner by telephone if the Examiner has any questions or suggestions concerning this amendment, the application, or allowance of any claims thereof.

I hereby certify under 37 CFR 1.8 that this correspondence is being facsimile transmitted to the United States Patent and Trademark Office on the date of signature shown below.
--

Respectfully submitted,
HUFFMAN PATENT GROUP, LLC

/ Richard K. Huffman /

By: _____

RICHARD K. HUFFMAN, P.E.
Registration No. 41,082
Tel: (719) 575-9998

03/14/2006

Date: _____